Muti-cycle Processor

Author	Lee Edith.D
Last Update	@March 15, 2022

Creating a Single Datapath from the Parts

- Assemble the datapath segments and add control lines and multiplexors as needed

What’s Wrong

- critical path:
  - `lw`
  - other leads to a waste

Multi-cycle Processor Implementation Overview

- Each instruction **step** takes 1 clock cycle
  - an inst. takes **more than 1** clock cycles to complete
- Allowance
  - faster clock rates
  - different inst. take a different number of clock cycles

    - functional units to be used more once per inst. as long as they are used on different clock cycles
        - only 1 memory
        - only 1 ALU/adder

High Level View

- Register needed after every major functional unit to hold the output value until it's used next
- only 1 instruction in datapath

Approach

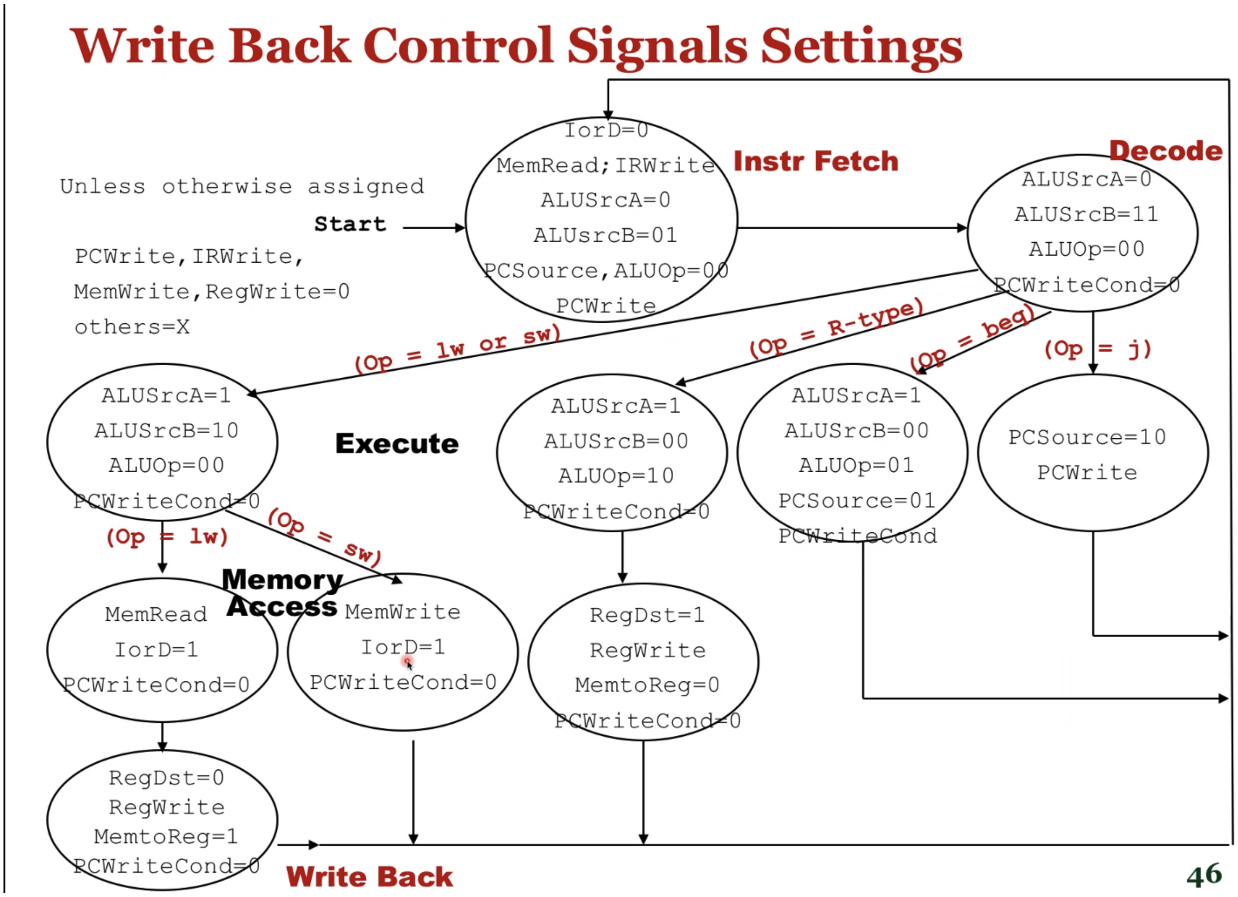
- Break up the instructions into many steps where each **step** takes a clock cycle while trying to
    - balance each step's work amount
    - use 1 functional unit per clock
- At the end of clock cycle
    - store values needed for next
        - IR
        - MDR
        - A and B
        - ALUout
    - no need for write control signal (except IR)
    - internal register is not visible
    - subsequent instruction data is visible

Steps

- Instruction Fetch
    - IR = Memory[PC]

    - PC = PC + 4
- Instruction Decode and Register Fetch
    - A = Reg[IR[25-21]]
    - B = Reg[IR[20-16]]
    - ALUOut = PC + (SignExt(IR[15-0])<<2)
- Execution for
    - R-type Instruction **Computation**
        - ALUOut = A op B
    - Memory Read/Write Address **Computation**
        - ALUOut = A + SignExt(IR[15-0])
    - Branch **Completion (3)**
        - if (A == B) PC = ALUOut
    - Jump **Completion (3)**
        - PC = PC[31-28] || (IR[25-0] << 2)
- Execution for
    - Memory Read/Write Access **Completion (4)**
        - MDR = Mem[ALUOut] `lw`
        - Mem[ALUOut] = B `sw`
    - R-type Instruction **Completion (4)**
        - Reg[IR[15-11]] = ALUOut
- Memory Read (Load) **Completion (5)**
    - Reg[IR[20-16]] = MDR

Control Signal Overview


Write Back Control Signals Settings

- input:
  - opcode
  - funct
  - **state(4-bit due to 10 states shown above overall)**
- output:
  - control signals in datapath
  - **next state**

CPI of Multi-cycle Processor

- assume each state in the CPU requires 1 clock cycle

Load	25%	5
Store	10%	4

Branch	11%	3
Jump	2%	3
ALU (R)	52%	4

  - $CPI = 0.25 \times 5 + 0.1 \times 4 + 0.52 \times 4 + 0.11 \times 3 + 0.02 \times 3 = 4.12$ (multi-cycle)
  - $CPI = 1$ (single-cycle)
  - $Performance = IC \times CPI \times Cycle\ time$
    - Multi-cycle Processor’s cycle time decreases
    - $Performace_{multi-cycle} < Performace_{single-cycle}$